**5th World Congress for Software Quality – Shanghai, China – November 2011**

# Success Factors to Achieve Excellent Quality

## - CMMI Level 5 Organizations Research Report -

Naomi Honda
NEC Corporation
Tokyo, Japan
E-mail: n-honda@ay.jp.nec.com

Key Words ： Software process, review, quantitative management, process quality, product quality, quality-centric software engineering culture, root-cause analysis, hands-on approach

**Abstract**

It remains important for a development organization to configure a software process that enables it to develop software with the least possible number of post-release defects. A development organization of CMMI level 5 has, over three years, been strived to improve those software products that had been noted as having many post-release defects. In this paper, it is reported on the following of the organization's improvement (Kaizen) activities, to analyze the success factors to achieve excellent quality. The results identified four important points; <1> early detection more than 80 % of defect during design or code review, <2> quality assurance for both side of process quality and product quality, <3> quantitative management with hands-on approach, and <4> quality-centric engineering culture and superior abilities for defect root-cause analysis.

## 1. Introduction

Software must take on the role of managing the systems that support the infrastructures on which society depends. Given this degree of dependence, society has the right to demand the development of software products that ship with as few post-release defects as possible. The venders that supply software products have difficulty in satisfying this requirement, however [1].

In response to the above, several different development and management techniques for improving the quality of software have been proposed. Among them, CMMI is widely applied worldwide. Unfortunately, even though a high level of CMMI is attained, satisfactory quality of software cannot always be obtained [2]. In fact, some development organizations with CMMI level 5 exhibit many post-release defects.

The author compared two organizations with CMMI level 5 and analyzed the key factors to achieve excellent quality [2]. The two organizations are called Organizations A and B. The two organizations apply almost the same software process and accomplished CMMI level 5 in the early 2000s. However, the number of post-release defects by organization B is over two times bigger than that appearing in the products of organization A. Organization B clearly had a quality problem with the number of post-release defects in its products. By analyzing the cause of the quality problem, the findings were attained as follows,

- Benchmarking using process data with other CMMI level 5 organizations has a CMMI level 5 organization find important ideas to achieve excellent quality.
- Superior abilities for defect root-cause analysis help to improve quality.
- Quality-centric software engineering culture is the foundation to perpetually maintain excellent quality. Quality-centric software engineering culture is based on the thinking way that quality is the highest priority in the organization and the culture has an affect on the behaviors of the developers in the organization.

The author has been driving the quality improvement activities (hereafter, this improvement activities call "Kaizen" activities) for three years in organization B and tracking the result. Based on the results, this paper discusses the success factors to achieve excellent quality. In addition, the appropriateness of the above three findings are reviewed. This paper discusses quality that focuses on having fewer defects after software shipment.

## 2. Overview of Organizations

This chapter outlines organizations A and B, and discusses the change of the number of post-release defects of the products produced by the organizations.

### 2.1 Overview of the Organizations

Organizations A and B belong to the same company and develop general-purpose IT-related software products in their different business areas. These organizations' customer groups are basically in the same enterprise area. Organization A has almost the same shipment volume, development amount, and number of engineers as B. The two organizations applied almost the same software process and accomplished CMMI level of 5 in the early 2000s. Usually, they would apply V model and implement V&V. Their development techniques, such as those for design and testing, are almost the same. Both adopt the "Quality accounting system™" internally created and developed as a quality management method [3].

### 2.2 Change of Number of Post-release defects before and after Kaizen

Figure 1 has histograms showing the distributions of the post-release defect for each product in Organizations A and B. Data for every product in Organizations A and B are summed up for one year. The data before Kaizen is one year before the start of Kaizen in organization B. The data after Kaizen is three years after the start of Kaizen in organization B. All numerical values are relative values, assuming the mean for organization A before Kaizen as 100. Smaller values indicate fewer post-release defects.

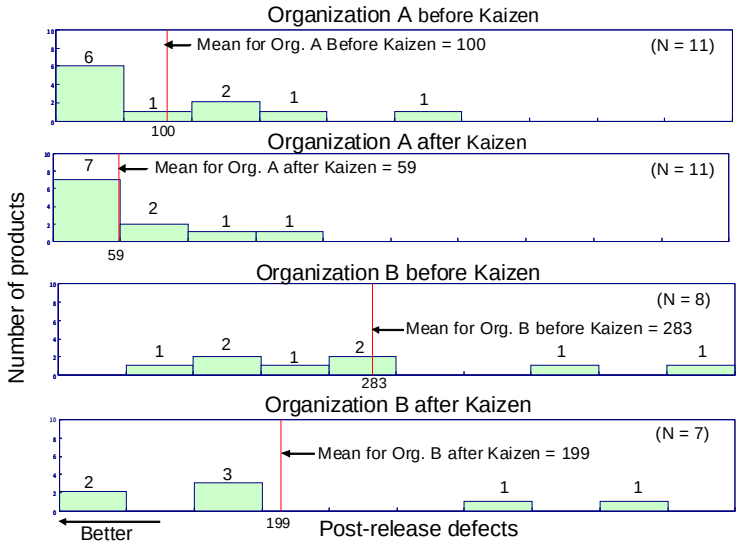


**Figure 1. Comparison of number of post-release defects**

When comparing these two organizations before Kaizen, it is clear that the values for Organization A before Kaizen are smaller. The mean for Organization A is 100 and that for Organization B before Kaizen is 283 (see organizations A and B before Kaizen). It can be shown that the number of post-release defects for B before Kaizen is over two times bigger than that for A before Kaizen. Organization B before Kaizen clearly had a problem with the number of post-release defects in its products. This triggered Kaizen activities in Organization B.

Next, it is focused to both after Kaizen data. The mean of post-release defects for Organization B before Kaizen was 283 but fell to 199 after Kaizen. Although the drop in Organization B was not sufficient to reach the same level as Organization A before Kaizen, the implementation of Kaizen did produce a significant reduction. The mean of post-release defects for Organization A after Kaizen was also decreased to 59. Organization A has been always running on their Kaizen activities and this is the effect of the Kaizen activities. This paper is discussed about focusing the results of organization B.

## 3. Benchmarking Organizations A and B

As discussed previously, benchmarking using process data with other CMMI level 5 organizations has a CMMI level 5 organization find important ideas to achieve excellent quality [2]. Using process data is especially essential to figure out the degree of difference between organizations. Process data is obtained in the course of software development such as efforts, number of defects, etc.

Organization B performed benchmarking with organization A from view points of process data and quality management system, and made Kaizen plans based on the benchmarking results. In these Kaizen plans, the plans that had good effects on the quality improvement are introduced below.

The data items used in this paper are listed in Table 1. Data on Organizations A and B before Kaizen and after Kaizen is listed in Table 2. Table 2 lists the relative values when each mean value in Organization A before Kaizen is 100 (hereinafter, all values are shown using these relative values). Data for every product in Organizations A and B are summed up for one year. The data before Kaizen is one year before the start of Kaizen in organization B. The data after Kaizen is three years after the start of Kaizen in organization B.

Table 1. Data items

| No. | Data item | Unit | Definition |
|---|---|---|---|
| 1 | Total effort | Person-hours/KL | Total person-hours needed to develop<br>Total effort = Design and coding effort (No. 2) + Review effort (No. 3) + Testing effort (No. 4) |
| 2 | Design and coding effort | Person-hours/KL | Person-hours needed for design and coding |
| 3 | Review effort | Person-hours/KL | Person-hours needed for design or code review |
| 4 | Testing effort | Person-hours/KL | Person-hours needed for testing |
| 5 | Total defect | Number of defects/KL | Total number of defects detected before shipment<br>Total defect = Defect during review (No. 6) + Defect during testing(No. 7) |
| 6 | Defect during review | Number of defects/KL | Of the Total defect (No. 5), the number of defects detected during upstream processes (design, coding and review) before the testing starts. Defects are mainly detected during design or code review in the upstream processes, so this number is called "Defect during review". |
| 7 | Defect during testing | Number of defects/KL | Of the Total defect (No. 5), the number of defects detected after testing starts. Defects are mainly detected by testing in the test process, so this number is called "Defect during testing". |
| 8 | Testing item | Number of test Items/KL | Number of testing items |
| 9 | Upstream defect detection rate | % | Ratio of the Defect during review(No.6) to Total defect (No. 5) |
| 10 | Success rate of 1+n procedure | % | "1+n procedure" is a quality improvement measure by detecting same-type defects if a defect is detected in the customer field. 1+n procedure performed where one or more same-type defects are detected is success.<br>Success rate of 1+n procedure = number of "1+n procedure" success cases / number of all "1+n procedure" performed cases |

Table 2. Descriptive statistics on the data

| No. | Data item | Organization A | | | | | | Organization B | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Before Kaizen | | | After Kaizen | | | Before Kaizen | | | After Kaizen | | |
| | | N | Mean | Std. Deviation | N | Mean | Std. Deviation | N | Mean | Std. Deviation | N | Mean | Std. Deviation |
| 1 | Total effort | 11 | 100.00 | 30.17 | 11 | **113.15** | **47.36** | 8 | 73.24 | 24.92 | 7 | 100.86 | 54.33 |
| 2 | Design and coding effort | 11 | 100.00 | 50.57 | 11 | **111.43** | **38.99** | 8 | 106.25 | 39.98 | 7 | 150.70 | 94.37 |
| 3 | Review effort | 11 | 100.00 | 31.22 | 11 | **91.24** | **23.71** | 8 | 47.56 | 9.17 | 7 | 81.48 | 22.91 |
| 4 | Testing effort | 11 | 100.00 | 33.79 | 11 | **120.12** | **74.45** | 8 | 54.69 | 22.57 | 7 | 67.35 | 39.45 |
| 5 | Total defect | 11 | 100.00 | 17.61 | 11 | **96.82** | **28.55** | 7 | 80.84 | 15.94 | 7 | 87.43 | 9.62 |
| 6 | Defect during review | 11 | 100.00 | 18.81 | 11 | **95.27** | **28.77** | 7 | 63.07 | 17.95 | 7 | 80.25 | 7.97 |
| 7 | Defect during testing | 11 | 100.00 | 18.31 | 11 | **106.21** | **61.65** | 7 | 187.77 | 46.48 | 7 | 130.68 | 34.30 |
| 8 | Test item | 11 | 100.00 | 34.56 | 11 | **114.66** | **82.14** | 8 | 57.82 | 22.05 | 7 | 129.75 | 69.64 |
| 9 | Upstream defect detection | 11 | 100.00 | 2.60 | 11 | **98.95** | **7.83** | 7 | 77.61 | 9.55 | 7 | 92.15 | 5.02 |
| 10 | Success rate of 1+n procedure | 11 | 100.00 | 66.77 | 9 | **73.09** | **66.49** | 8 | 30.49 | 61.23 | 7 | 81.06 | 53.47 |

**Note. All numerical value are relative values, assuming the mean for organization A before Kaizen as 100.**

### 3.1 Benchmarking using process data

In this chapter, it is discussed about the results of benchmarking organization A before Kaizen and organization B before Kaizen using table 2.

As shown in table 2, there are four data items these mean values are less than 60. These are review effort, testing effort, testing item and success rate of 1+n procedure (see data items Nos. 3, 4, 8, 10 in table 2). These data items except success rate of 1+n procedure are discussed first. Then success rate of 1+n procedure is discussed next.

These three data items, they are review effort, testing effort and testing item, also indicate amount of quality checking. These mean values of three data items show that amount of quality checking for B is quite low and about half of those for A. In contrast, there is only defect during testing for organization B which mean value is bigger than over 40 that for organization A (see data item No. 7 in figure 2). Organization B indicates that defect during review is low and defect during testing is high. On the other hand, organization A indicates opposite tendency comparing with organization B.

Organization A has been driving Kaizen activity to detect defects during design or code review over twenty years. As a result, the value of upstream defect detection rate improved over 80 percent. Upstream defect detection rate is the ratio of the defects during design or code review to total defects before shipment. By this kaizen activity, the post-release defects of organizations A was reduced one twentieth twenty years ago. The success history of organization A indicates the improvement of defect detection during review works to reduce post-release defects. Based on this experience and above results, organization B decided to reinforce review activity as kaizen measures.

Then, success rate of 1+n procedure is considered, which mean value for organization B is less than over 40 that for A. “1+n procedure” is one of our preventive actions for shipped products, and is to detect same-type defects if a post-release defect is detected in the customer field (hereafter, post-release defects detected in the customer field call customer defects). Meaning of the word “1+n procedure” is that “1” refers to one customer defect and “n” refers to the number of same-type defects. One same-type defect is a defect remaining in the shipped product that has the same root-cause as the customer defect. The 1+n procedure refers to detecting same-type defects by analyzing the root-cause for the customer defect. Therefore, the ability for defect root-cause analysis is superior, so more same-type defects can be detected by the 1+n procedure. When one or more same-type defects are detected by the “1+n procedure”, the “1+n procedure” case is regarded as success, on the other hand, no same-type defects detected is regarded as failure. The Success rate of 1+n procedure is the ratio of number of “1+n procedure” success cases to number of all “1+n procedure” performed cases. A higher Success rate of 1+n procedure means to keep higher ability of defect root-cause analysis. Same-type defects detected by the “1+n procedure” are corrected to the shipped product as corrective actions. Therefore, a higher success rate of 1+n procedure is directly effective to reduce post-release defects. In these two organizations, the developer is obliged to perform “1+n procedure” to all customer defects.

The mean value of success rate of 1+n procedure for organization B is only one-third of that for A (see data item No. 10 in table 2). Improving success rate of 1+n procedure must be helped to improve the ability of defect root-cause analysis and to reduce post-release defects in organization B. Based on that analysis, organization B decided to improve success rate of 1+n procedure as kaizen measures.

### 3.2 Benchmarking Quality Management system

Organizations A and B both have Quality Assurance groups (hereafter, called QA groups) that are independent of the development group. The QA groups in both organizations monitor quality problems through the start of software development to shipment and urge the development group to take corrective action for any problems as needed. There is a difference, however, in the details of the monitoring and actions in the two organizations.

Organization A checks product quality on a weekly basis, using weekly data obtained in the course of software development, while conducting an evaluation of the final products based on customer perspectives (hereafter, called independent QA testing) that is independent of the testing performed by the development group. The organization holds a weekly project management meeting that is attended by the managers of the development group and the QA group. In this meeting, they identify development problems

and solve them based on the results of analyzing the process data obtained during the course of the development. At the end of the development, they determine whether a final product can be shipped. If independent QA testing reveals a great number of defects with the product, then the product would likely lead to frequent defects once in the hands of a customer, such that it would be classed as unacceptable for shipment and shipments would be delayed. The managers of the QA group and the development group both have the right to determine whether a product can ship.

On the other hand, Organization B presents the following points that differ from those of Organization A.

- Organization B does not conduct independent QA testing.
- The QA group of the organization B checks the quality of the product only upon the completion of each process, not weekly basis. And the way of checking in organization B is in writing.

The former is discussed. The purpose of independent QA testing is to actually run the final software product that will be delivered to customers and evaluate it from the customers' perspectives. Consequently, this test enables us to actually detect any defect in the software that cannot be identified by quality monitoring based on data obtained during the development process.

Next, the latter is discussed. It should be executed data collection and quality checking on short term with face to face communication. Because, completion of each process basis would be on long term, such as one month or more, and it might be delayed to figure out problems occurred in the middle of development process. In addition, if it could be figured out the problems, the degree of its seriousness might be misunderstood because there was no data. And face to face communication based on project management meeting can be detected development problems easier and earlier than in writing.

Based on the results of the above benchmarking, it was decided that the following Kaizen measures should be implemented to reinforce the quality management system in organization B.

- Implementation of independent QA testing of the final product
- Data collection and quality checking at project management meeting on a weekly basis

## 4. Results of Kaizen

The results of the number of post-release defects are shown in chapter 2.2. In this chapter, the results of each Kaizen activity in organization B are discussed.

### 4.1 Results of Reinforced Review Activity and improved success rate of 1+n procedure

The post-Kaizen data for Organization B is listed in Table 2 (see Organization B after Kaizen). Figure 2 shows the comparison of mean for organization B before versus after Kaizen. All data items except defect during testing were increased. The value of review efforts increased to almost double of previous value under this review enforcement campaign (see Data item No. 3 in figure 2). At the same time, the value of defects during review increased 27 percent from 63.07 before Kaizen to 80.25 after Kaizen (see data item No. 6 in table 2). The aim of the review enforcement campaign was to increase defects during review by reinforcing review activity. Therefore, the review enforcement campaign got success and had the aimed results.

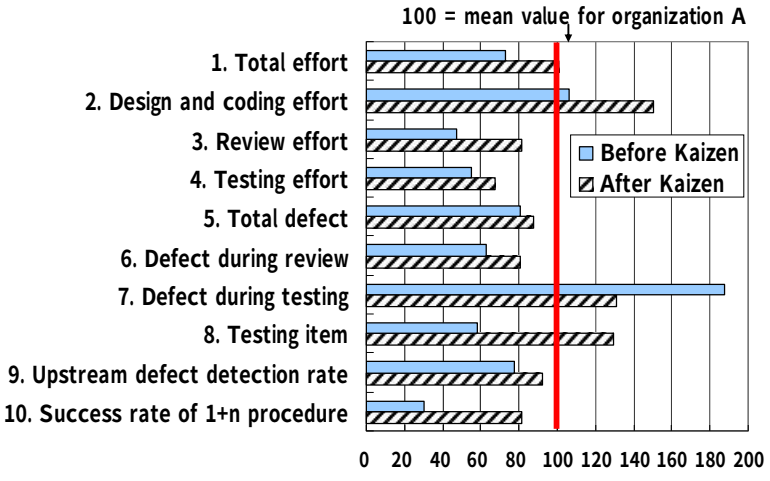


**Figure 2. comparison of mean for organization B (Before vs. after Kaizen)**

Organization B also tried to improve testing. But the number of testing efforts increased modestly and the number of

testing items increased almost double of previous value (see data items Nos.4 and 8 in figure 2). In contrast, the number of defects during test decreased 44 percent from 187.77 before Kaizen to 130.68 after Kaizen (see data item No. 7 in table 2). At first glance, it looks to seem that the efficiency of defects detection during testing turned worse, because the number of testing items increased almost double but the number of defects during test decreased. However, the number of total defects increased 8 percent (80.84 before Kaizen to 87.43 after Kaizen (see data item No. 5 in table 2)). According to our analysis, as a result of improving review, most defects were successfully detected by review, with fewer defects remaining in the software at the start of testing. For this reason, improved testing detected fewer defects than before Kaizen.

Based on the above, the Kaizen measure of the reinforced review activity enables organization B to ensure quality by defect detection during design or code review. This means that organization B is becoming to be able to ensure quality in earlier upstream process.

Next, the number of success rate of 1+n procedure for organization B was increased to more than double of previous number, from 30.49 before Kaizen to 81.06 after Kaizen (see data item No. 10 in table 2). And now, that for organization B became close to that for A.

### 4.2 Results of changed the quality management system

Organization B changed the quality management system under the Kaizen measures. The results of implementation of independent QA testing of the final product are discussed first. Then results of data collection and quality checking at project management meeting on a weekly basis are discussed next.

The implementation of independent QA testing was very effective. Independent QA testing for all developed products was implemented by the QA group. As a result, about 3% of the total defects were detected by the independent QA testing. The development group whose product defect was detected by independent QA testing conducted additional testing and detected 1 % of the total defects. Consequently, it can be said that independent QA testing enabled the extraction of 4 % of the total defects, in comparison with the detection rate before Kaizen.

Next, data collection and quality checking at project management meeting on a weekly basis became to be able to figure out problems clearly through the whole development processes. For instance, the organization B began to set standard values for the review effort, and checked the degree to which these are achieved during the software development process. This improvement enabled it to identify the development group when it has varied from the standard values, and to eliminate the need for follow-up activities that would otherwise be required if any unsatisfied conditions were found after a project. This would be the second reason for the success of the review enforcement campaign.

## 5. Discussion

Based on the Kaizen and its results in Organization B, this chapter discusses the success factors to achieve excellent quality, and the appropriateness of the findings obtained in the past are reviewed.

### 5.1 Effect of Reinforced Review on the Number of Post-release defects

In general, review is important and an efficient activities of ensuring the quality of a software product during its development [5]. It is used upstream defect detection rate as a data item explained about review effects in organizations A and B. As described in chapter 3.1, organization A keeps the mean of upstream defect detection rate is over 80 percent and there are very few products which value of upstream defect detection rate is less than 80 percent. To reach and keep 80 percent, Organization A internally created and has been applying “Quality accounting system™” [3]. That’s why organization A keeps that there are very few defective products after shipment.

The mean of upstream defect detection rate for organization B before Kaizen was less than 70 percent and there was no product which value of upstream defect detection rate was more than 80 percent. Then three years later, by these kaizen activities including “Quality accounting system™” applying accurately, that for organization B after Kaizen was increased and was very closed to 80 percent. There are three products in organization B after Kaizen which value of upstream defect detection rate is more than 80 percent. It’s 43

percent of the total number of products in organization B. And the mean of post-release defects for organization B has been decreasing.

It is said that early ensured quality by defect detection during review is important for good quality. These above results have it take another step forward, that is, early detection more than 80 % of defect during review is a key to achieve excellent quality.

### 5.2 Effect of Quality Assurance Mechanism from both side of process quality and product quality

An input for quality management is the importance of quality assurance for both the process quality and product quality [6]. Organization A monitors the implementation of a development process based on data for a software product that is under development, and performs quality assurance for the final product through independent QA testing.

On the other hand, before Kaizen, Organization B only conducted process quality assurance and lacked any kind of product quality assurance. The adoption of independent QA testing allowed for the functions of product quality assurance. More importantly, due to the fact that independent QA testing actually detected defects in a software product, an excellent understanding about Kaizen was gained by those engineers who were originally skeptical about the activity. This suggests the importance of demonstrating any issue and must have acted as a spur to instigating the Kaizen activities in Organization B.

These results seem to show that a proper combination of process and product quality assurance methods is a key to the development of a software product with excellent quality.

### 5.3 Effect of Quantitative management

Twelve Japanese organizations that had accomplished CMMI level 5 were surveyed [7]. According to the results of a questionnaire survey, the greatest advantage of accomplishing this level is the establishment of quantitative management. This survey seems to give suggestions of the importance of the hands-on approach. The hands-on approach means that people visit the location of the trouble, look at the actual objects there, and observe what is really happening, instead of sitting at their desks theorizing [8]. The hands-on approach is necessary to have an effect on software development. The quantitative management required by CMMI level 5 organizations must to emphasize real development site to have good effects. In fact, the reason why that organization A has been always running on having an effect on their Kaizen activities is that organization A is considered the hands-on approach is important [3].

In Organization B, the Kaizen was implemented to collect weekly data and to hold project management meetings. This improvement enabled timely and careful analysis and follow-ups during development. Therefore, the problems were solved at early times and it was begun to work PDCA cycle that is to reconsider the software process from view points of the problems in order to the next success. In addition, it was actually felt that became to communicate smoothly with persons concerned the projects because of a weekly basis follow-up meeting. These circumstances seem to change from sitting at desks theorizing to emphasis of actual world.

Hence, quantitative management with hands-on approach is a key to achieve excellent quality.

### 5.4 Root-cause analysis and Quality-centric software engineering culture

Among the findings obtained in the past, the abilities for defect root-cause analysis and quality-centric software engineering culture are discussed in this chapter.

First, it was discussed about the relationship between abilities for defect root-cause analysis and quality improvement. Analyzing root-cause of defects requires exact understandings to the defects. That is, the improvement of abilities for defect root-cause analysis encourages improvement of abilities for understandings. And the improvement of these abilities has good effects of all kinds of development works. It is thought that the reasons for the success of the review enforcement campaign are the improvement of accuracy for defects detection during review using the abilities for root-cause analysis, not only increase of review efforts. As above discussion, it seems that improvement of the abilities for root-cause analysis leads to improve quality, but not directly.

Next, it was discussed about the relationship between quality-centric software engineering culture and quality improvement. There were behavioral changes of engineers in organization B for three years. Three

years ago, there were very few words of participants in quality meeting. However, lively discussion is made now in the same meeting. These changes were also observed behavior pattern of top management. Three years ago, the top managements in organization B were never mentioned about their product quality policy to their employees in their address at the beginning of the year. However after three years, the same top managements are stated about the importance of the product quality and the quality target in the same stage. And top management themselves held a quality enhancement event which aimed to discuss their quality problems. Based on these changing, it can be said that it is now building up quality-centric software engineering culture in the organization B. In the above CMMI level 5 Organization survey, it was pointed that one of the most essential factors to improve quality was “a quality-centric software engineering culture” [7]. It can be said lots of organizations feel the importance of quality-centric software engineering culture. As above discussion, it seems that a quality-centric software engineering culture has a strong connection with quality improvement.

## 6. Conclusions

Focusing on the reduction of post-release defects, this paper analyzed the application of Kaizen in organizations of the CMMI level 5, to discuss the characteristics of the software process for developing a software product with excellent quality.

According to the findings given in this paper, the following four points should be taken into account when developing a software product with excellent quality.

- Early detection more than 80 % of defect during design or code review
- Quality assurance for both side of process quality and product quality
- Quantitative management with hands-on approach
- Quality-centric software engineering culture and superior abilities for defect root-cause analysis are important keys to improve quality

There are many organizations that try to improve their software processes to improve the quality of their software products. The points mentioned above for the reduction of post-release defects are grasped to construct a software process. This can be effective for realizing a reduction in the number of the post-release defects.

In the future, the author will continue to study the software processes needed for the development of excellent quality software.

**Acknowledgments**

The author would like to express my deep gratitude to Mr. Yamamoto, Operating Officer, NEC Corporation for his great cooperation.

**References**

[1] M. A. Cusumano, “The Business of Software”, New York : Free Press, 2004.

[2] N. Honda, “Beyond CMMI Level 5 - Comparative Analysis of Two CMMI Level 5 Organizations -“, Software Quality Professional, vol. 11, no. 4, pp.4-12, 2009

[3] N. Honda, “Software Quality Accounting System – Quality assurance technology that supports high quality software development at NEC”, JUSE, 2010

[4] B. Boehm, “Software Engineering Economics”, Prentice-Hall, 1981

[5] T. Gilb, and D. Graham, “Software Inspection”, Kyoritsu-press, 1999

[6] SQuBOK study team, “SQuBOK Guide – Software Quality Body of Knowledge”, Ohmsya, 1997

[7] JUSE, “Research report for CMMI level 5 organizations”, 2009

[8] T. Ohno, “Toyota Production System”, Diamond-press, 1978